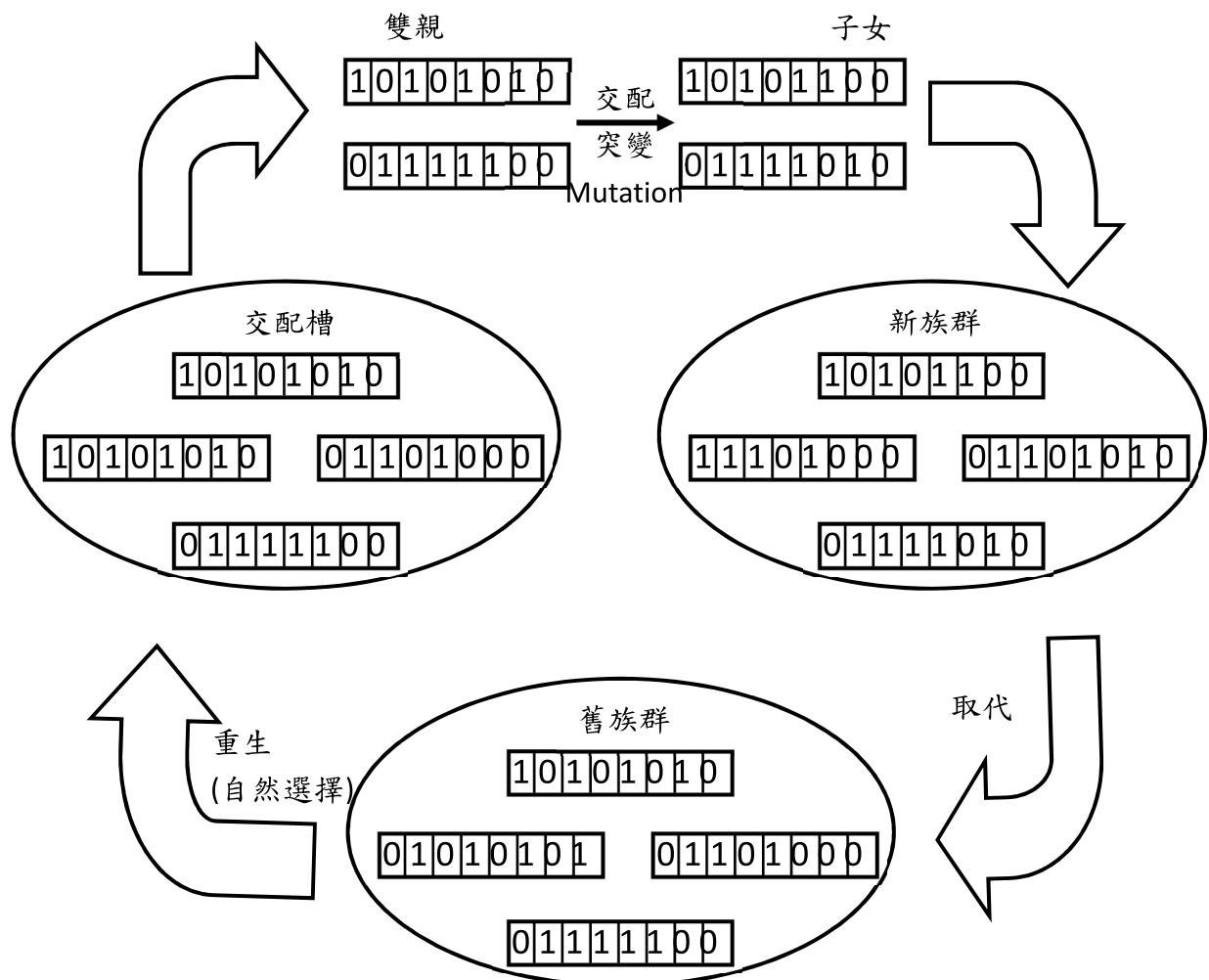


一. 採用的方法介紹

基因演算法的架構流程包括：編碼與解碼結構、初始族群、適應函數之評估、母代選擇、交配運算子、突變運算子。

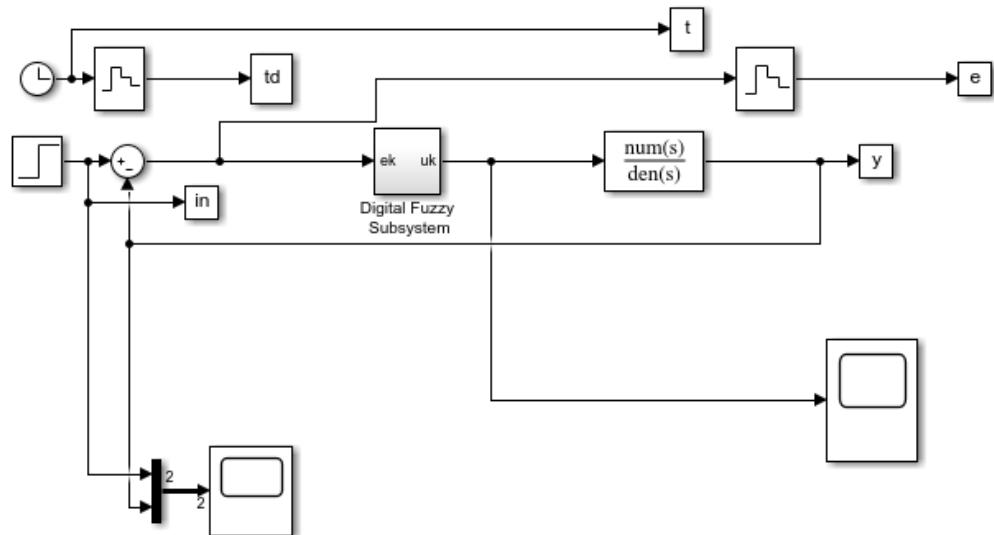
1. 染色體：將參數空間中沒中可能的解進行二進位子串編碼，這個編碼的值稱為染色體 (chromosome [\kromə,som])。
2. 基因 (gene)：組成母代的個體稱為基因。
3. 族群 (population)：在基因演算法中將基因之集合稱為「族群」，或稱之為「基因池 (gene pool) 」。
4. 世代 (generaion)：演算法中每隔一段時間透過基因運算程序如：天澤、交配與突變所產生的新的族群，次時間週期數稱作「世代」。

遺傳演算法的演化過程

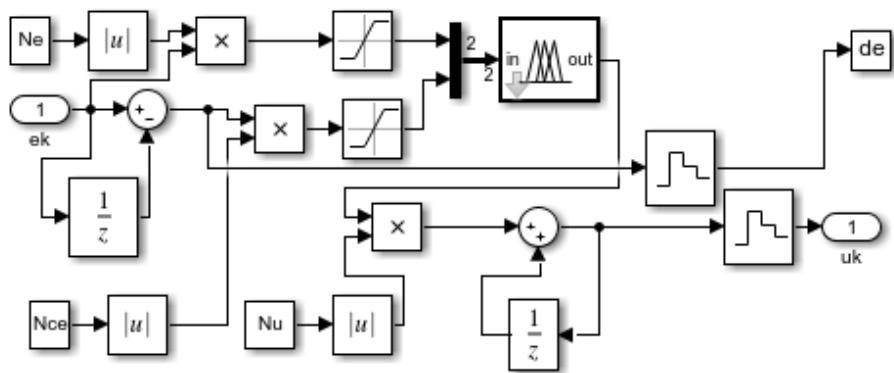


二. 所需要解決問題

K=學號後二位數(例如: U041270**01**)



$$\frac{num(s)}{den(s)} = \frac{K}{s^2 + 5s + 6}$$



找出適當的 Ne Nce Nu ，使系統的輸出滿足下列條件

1. 最大超越量小於 30%
2. 安定時間小於 3 秒(**$\pm 5\%$**)
3. 穩態誤差為零

三. MATLAB&SIMULINK(所提供檔案)

```
gfc199A_yen.slx=>所需要解決的問題  
main.m  
clear all  
gfc=readfis('gfc');%代入自己設計得gfc.fis檔案  
global Ne Nce Nu Ts;  
Ne=1;Nce=1;Nu=1;Ts=0.5;  
K=1;%學號後二位數(例如:U04127001)
```

採用 GA 找尋三個參數

採用 PSO 找尋三個參數

Or

%需要自己完成

提示：可以採用MATLAB function : ga or particleswarm 需要搭配optimoptions設定相關參數

optimoptions設定參數: size設定為10，最多迭代50次，畫出每次迭代狀況

GA: 'PopulationSize',10, 'Generations',50,'PlotFcn', 'gaplotbestf'

PSO: 'SwarmSize',10,'MaxIterations',50,'PlotFcn', 'pswplotbestf'

fintness.m

```
function fitnessvalue=fintness(x)  
global Ne Nce Nu Ts;  
mdl='gfc199A_yen.slx';  
load_system(mdl);  
assignin('base','Ne',x(1));  
assignin('base','Nce',x(2));  
assignin('base','Nu',x(3));  
sim(mdl,10); %執行10秒鐘  
stepvalue = 1;  
[OverShoot, AdjustTime, SteadyStateError] = Performance(t, y, stepvalue); %需  
要自己完成
```

自我撰寫出一個 performance 的 function，輸出分別為最大超越量，安定時間與穩態誤差。安定時間定義為進入與穩態差異 $\pm 5\%$

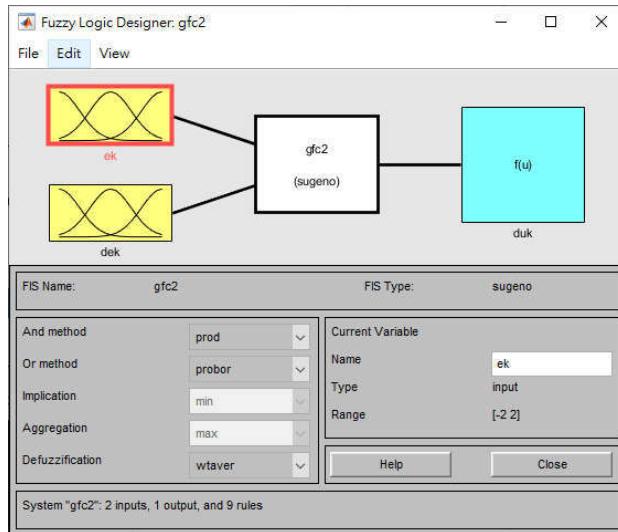
fitnessvalue=?%需要自己完成，需要滿足下列條件

1. 最大超越量小於 30%
2. 安定時間小於 3 秒
3. 穩態誤差為零

四. 如何設計 FIS 架構，請說明

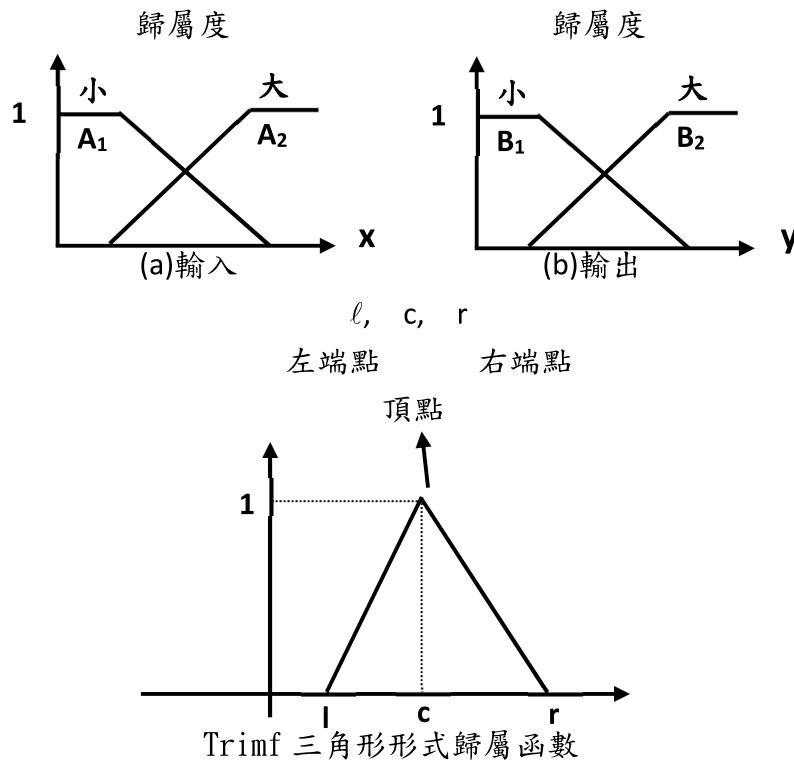
簡述如何設計出 FIS 架構，且放上 FIS 輸入、輸出和 Rule 的規則

Fuzzy 工具打開如下，需定義輸出輸入的歸屬函數以及判斷規則：

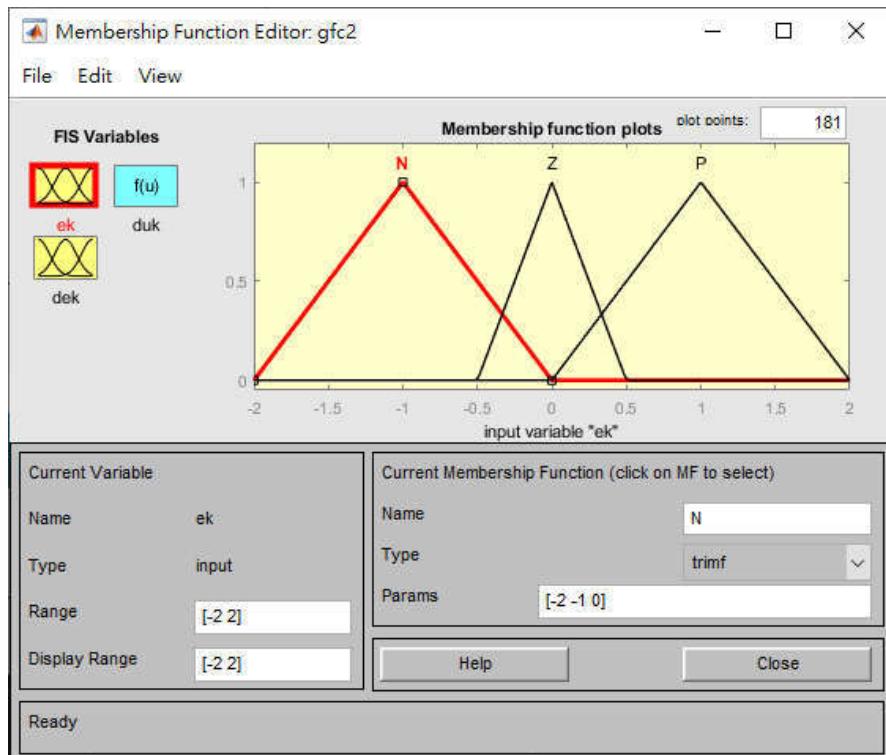


Fuzzy Logic Designer 外觀

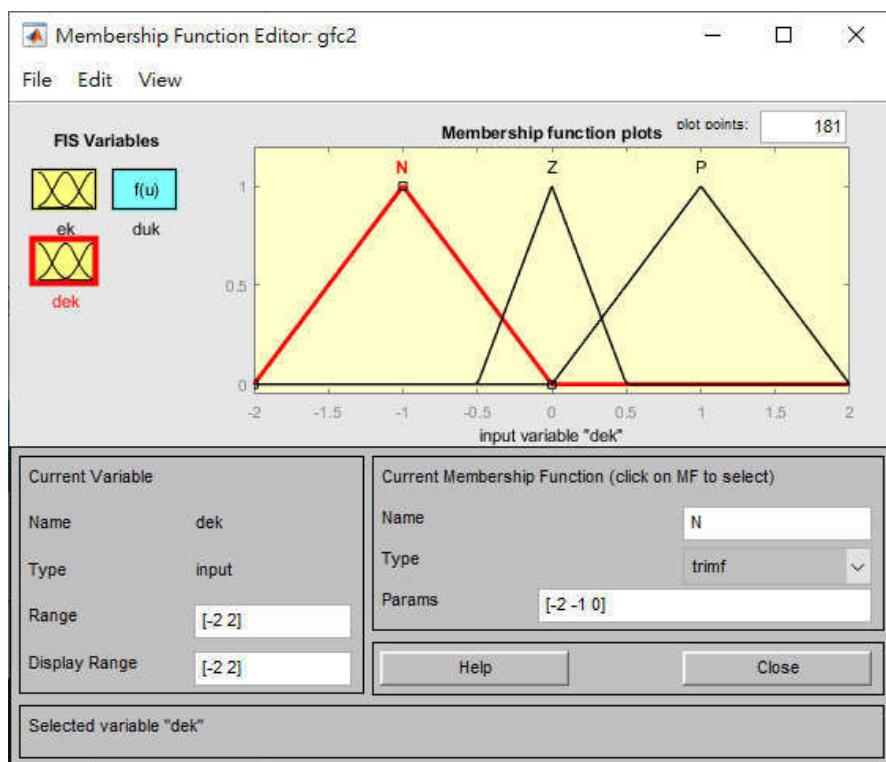
設定兩個歸屬函數 ek 、 dek ，並且使用 trimf type 的歸屬函數形式
大概類似於有下方的三角形，用三個函數定義它。



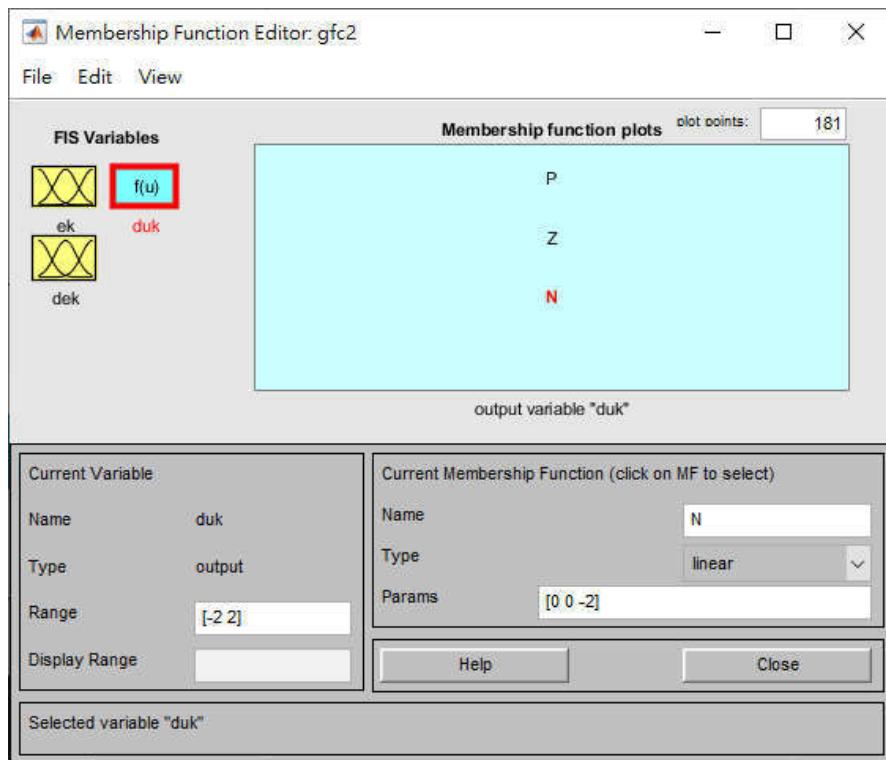
下圖分別為 ek、dek 以及輸出的歸屬函數：



ek 值歸屬函數



dek 值歸屬函數

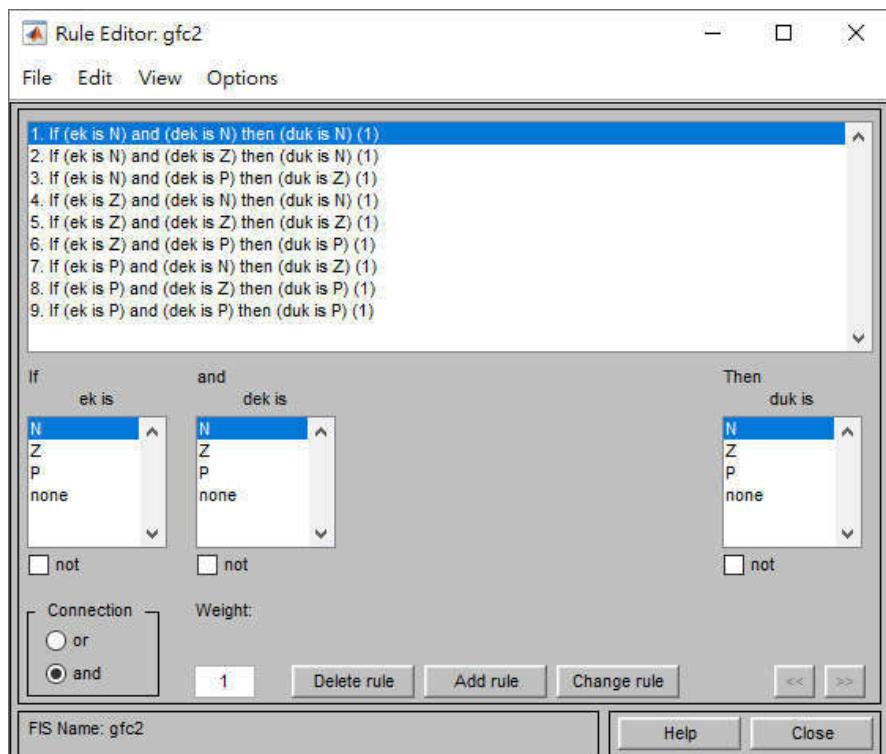


duk 輸出歸屬函數

規則設定：一般的模糊邏輯系統，都是先有規則才開始設計的。

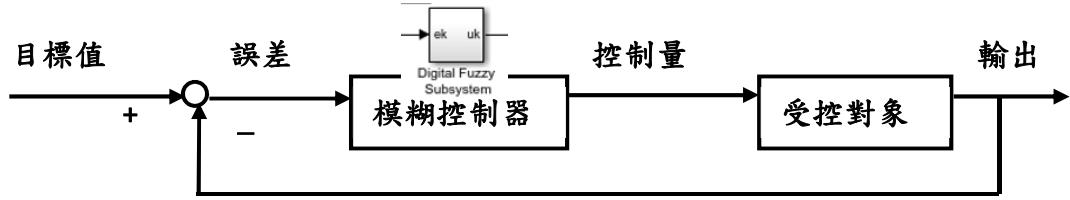
比如我們的系統就要有規則如下：

ek、dek 小則輸出小，ek 大、dek 大則輸出大.....



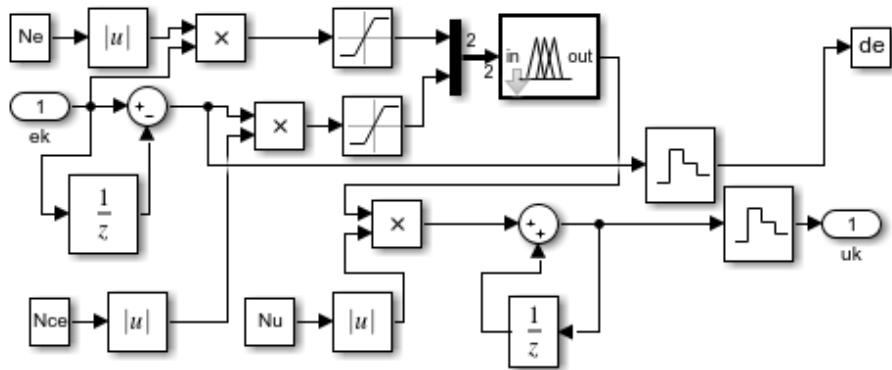
規則設定

以我們的系統為例，主要架構如下：



使用固定歸屬函數，調整模糊規則（sugeno 模糊系統）

模糊控制系統架構：



模糊控制系統架構

$$\begin{aligned} \text{Input of Fuzzy Designer} = & \begin{cases} e_k = e_k \times |Ne| \\ de_k = (e_k - e_{k-1}) \times |Nce|, > 0 \text{ 上升}, < 0 \text{ 下降} \end{cases} \\ u_k = & (du_k \times |Nu|) + (du_k \times |Nu|)_{k-1} \end{aligned}$$

從上面得知大概設計的規則，套用到我們的系統之中。

假設 Ne 、 Nce 、 $Nu = 1$ ，則 $N = \text{negative}$ 、 $Z = \text{zero}$ 、 $P = \text{Positive}$

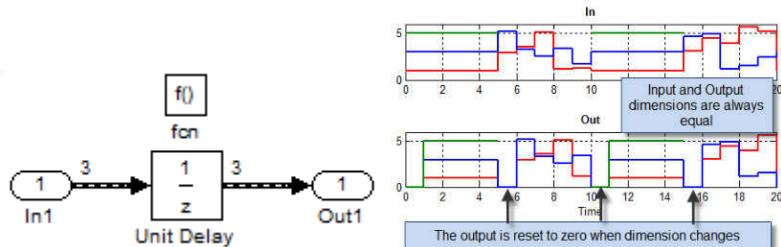
$e_k \backslash de_k$	N	Z	P
N	N	N	Z
Z	N	Z	P
P	Z	P	P

若 de_k 歸屬於 N (與前一訊號相比下降)， $e_k = N, Z, P$ ，輸出 N、N、Z

若 de_k 歸屬於 Z (與前一訊號相比不變)， $e_k = N, Z, P$ ，輸出 N、Z、P，與 e_k 相同。

若 de_k 歸屬於 P (與前一訊號相比上升)， $e_k = N, Z, P$ ，輸出 N、P、P

unit delay ($\frac{1}{z}$) 的作用就是將數據延時一個採樣時刻。它的輸入經過一個採樣時刻的延時，變成輸出。如果要用前一個數值減去後一個數值除時間來速度，輸出為上一個採樣時間數值，將二者信號線抽出來做減法，再除以長度即可。例如：



五. 自己設計的相關程式，請說明

放入你所撰寫的三個程式內容，並加上註解

1. main.m
2. fintness.m
3. Performance.m

1. main.m

```
clear all
clc
warning('off','all')
%{
開始設定初值
Ne=1;Nce=1;Nu=1;Ts=0.5;
二階系統 num(s)/den(s) = K / (s^2+5s+6), K = 32
%}
gfc=readfis('gfc'); %請自己設計gfc.fis
global Ne Nce Nu Ts;
Ne=1;Nce=2;Nu=3;Ts=0.5;
K=32; %學號後二位數(例如: U04127001)

%{
演算法設置 (optimptions) 在學校電腦，設置無法使用！？
'...' 換行的意思
#GA: 'PopulationSize',10, 'Generations',50,'PlotFcn', 'gaplotbestf'
'Generations',50 ... % 迭代次數
'PopulationSize', 60, ... % 染色體數量
'MaxGenerations', 100, ... % 最大繁衍代數
'PlotFcn', {@gaplotbestf, @my_plot}, ... % 繪圖函數 @my_plot可以自定義，也可以不寫
    1. @gaplotbestf %最佳分數及平均分數
    2. @gaplotscores %每代個體的分數
    3. @gaplotdistance %每代個體間的平均距離
    4. @gaplotscorediversity %繪製每一代分數的直方圖
'CrossoverFraction', 0.8, ... % 交配率
'MutationFcn', {@mutationuniform, 0.1}, ... % 突變率
'Display', 'iter'); % 結果展示方式
```

```

#PSO: 'SwarmSize',10,'MaxIterations',50,'PlotFcn','pswplotbestf'
'SwarmSize',10 ... % 群體數量
'MaxIterations',50 ... % 迭代次數
'PlotFcn', {@gaplotbestf}, ... % 繪圖函數
'pswplotbestf'
%}

GA_options =
optimoptions('ga','PopulationSize',10,'Generations',50,'PlotFcn',{@gaplotbestf});
PSO_options =
optimoptions('particleswarm','SwarmSize',10,'MaxIterations',50,'PlotFcn','pswplotbe
stf');

```

%{

用下面程式或是打開 optimization tool 的 ga

找尋Ne、Nce、Nu最佳值

```
[x, fval] = ga(@fintness, nvars, A, B, C, D, LB, UB, [ ], IntCon, GA_options)
[最佳解, 適應值] = ga(適應函數, 變數數量, A, B, C, D, 下界(-2), 上界(2), 非線性約束
式, 整數欄位, 參數)
```

其中，限制式為： $A*x < B$ 、 $C*x = D$ 。

$A*x' - b$

%{

A = [];

b = [];

%Aeq = [-1 1];

%beq = 5;

fun = @fintness;

```
[x, fval] = ga(fun,3,A,b,[],[],-2,2,[],[],GA_options);
```

disp(A*x' - b);

%}

2. fintness.m

```
function fval=fintness(x)
%{將GA或是PSO得到的最佳解代入 x(Ne,Nce,Nu)
並求得計算響應的指標（最大超越量小於30%、安定時間=3s 容忍±5%、穩態誤差=0）
F(x) = | Kmax - F(x) --- PO<30% 、s <= 3s 、error = 0
| 0
%}
global Ne Nce Nu Ts;
mdl='gfc199A_yen.mdl';
load_system(mdl);
assignin('base','Ne',x(1)); % assignin 重新給定數值
assignin('base','Nce',x(2));
assignin('base','Nu',x(3));
sim(mdl,50); % 重新跑模擬一次
%%計算響應的指標
[OverShoot, AdjustTime, SteadyStateError] = Performance(t, y, control);
%fval 小的時候收斂，越小越接近最佳解
if OverShoot < 30
    os = 0;
else
    os = 1;
end
if AdjustTime < 3
    tst = 0;
else
    tst = 1;
end
if SteadyStateError <= 0 % 不可能為零，放在 LSB
    er = 0;
else
    er =1;
end
fval = os*100+tst*10+er; % 三個bit的二進位數，oas (OS、settling time、error)
fprintf('\n fitnessVal(oas): %d', fval);
fprintf('\n OverShoot: %d \n AdjustTime: %d \n SteadyStateError: %d \n', ...
    OverShoot, AdjustTime, SteadyStateError);
fprintf(' Ne: %d \n Nce: %d \n Nu: %d \n', x(1), x(2), x(3));
```

3. Performance.m

```
function [OverShoot, AdjustTime, SteadyStateError] = Performance(t, y, control);
%{ 理論部分
    找二階系統 num(s)/den(s) = K / (s^2+5s+6), K = 32
    0< $\zeta$ <1,  $\omega_n$  > 0 : 控制訊號 G(s) =  $\omega_n^2$  / (s(s+2 $\zeta$  $\omega_n$ ))
    OverShoot (P.O.) 最大超越量 = (最大值 - 穩態值) / 穩態值
    AdjustTime (ts) 安定時間 = ln(tolerance fraction)/( $\zeta$   $\omega_n$ )
    二階欠阻尼系統的阻尼比  $\zeta$  << 1,  $\zeta$  使用0.7, 容忍誤差±5%, ts = 3s
    4* 3s = ln(0.05) / (0.7 $\omega_n$ ) =>  $\omega_n$  = 0.3566
    omega_n = abs(log(0.05))/(4*3*0.7) => 0.3566
    AdjustTime_ts = abs(log(0.05))/(0.7*0.3566*4) => 3.0003
    SteadyStateError = 穩態值 - 控制輸入
    終值又被稱為穩態值(Steady state value),
    穩定輸出與控制輸入的差值定義為穩態誤差ess(Steady state error)
    步階輸入之ess = 1 / ( 1 + lim(s->0)[G(s)] ) = 1 / (1+Kp)
}

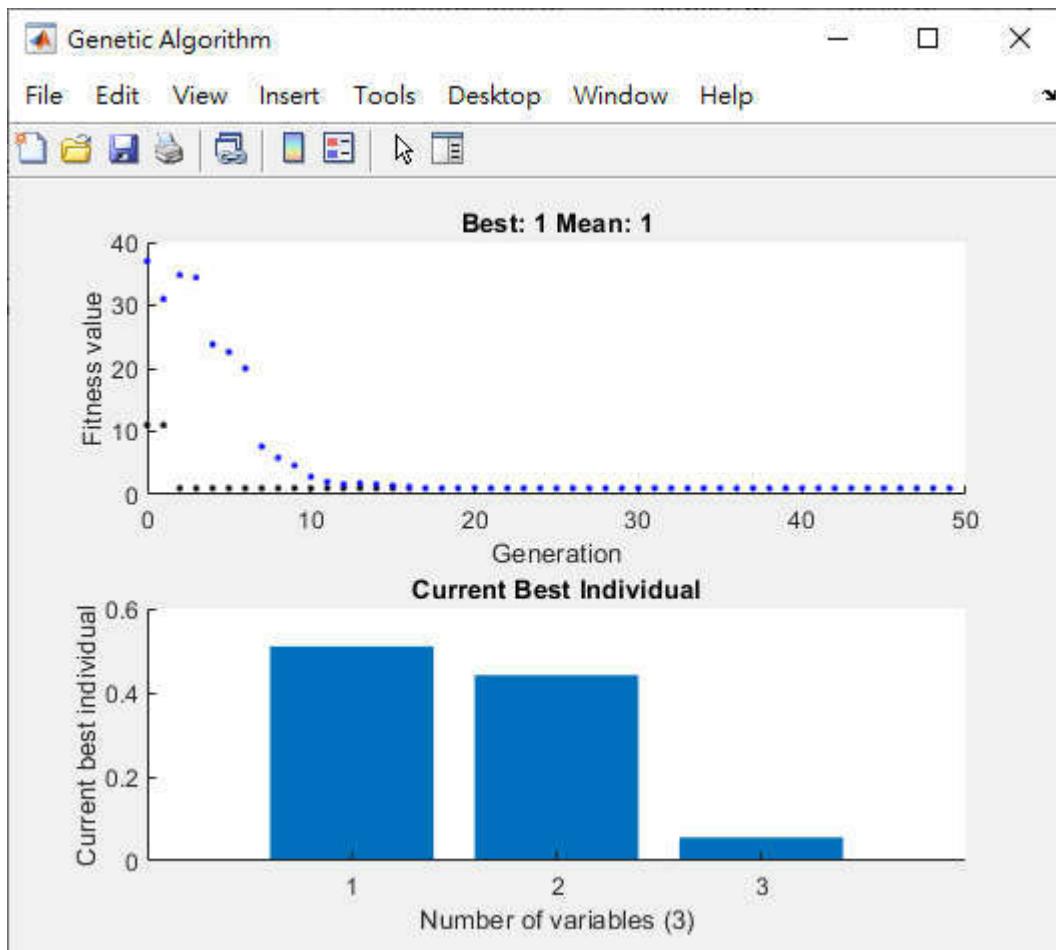
%}
%omega_n = abs(log(0.05))/(4*3*0.7);
%OverShoot = (max(y) - y(end))/y(end)*100;
%AdjustTime = abs(log(0.05))/(0.7*0.3566*4);
%SteadyStateError = 1/(1 + control(1));
S = stepinfo(y);
oss = S.Overshoot;
tas = S.SettlingTime;
ers = 1/(1 + control(1));

Static = find((y>1*1.02)&(y<1*0.98),1,'last');
AdjustTime = t(Static);
OverShoot = (max(y)-mean(y(Static:end)))/mean(y(Static:end))*100;
SteadyStateError = abs(1-mean(y(Static:end)));
fprintf(' QAdjustTime(ta): %d \n QOverShot(os): %d \n QError(er): %d \n',
AdjustTime, OverShoot, SteadyStateError);
fprintf(' AdjustTime(ta): %d \n OverShot(os): %d \n Error(er): %d \n', tas, oss,
ers);
```

六. 結果

1. 畫出每次迭代狀況
2. 最佳的三個 N_e N_{ce} N_u 為何，最大超越量、安定時間與穩態誤差
3. 兩張圖
 - i. 輸出輸入同一張圖
 - ii. 控制訊號

1. 畫出每次迭代狀況



2. 最佳的三個 N_e N_{ce} N_u 為何，最大超越量、安定時間與穩態誤差

fitnessVal(oas): 1

OverShoot: 1.098088e-01

AdjustTime: 2

SteadyStateError: 1.096884e-03

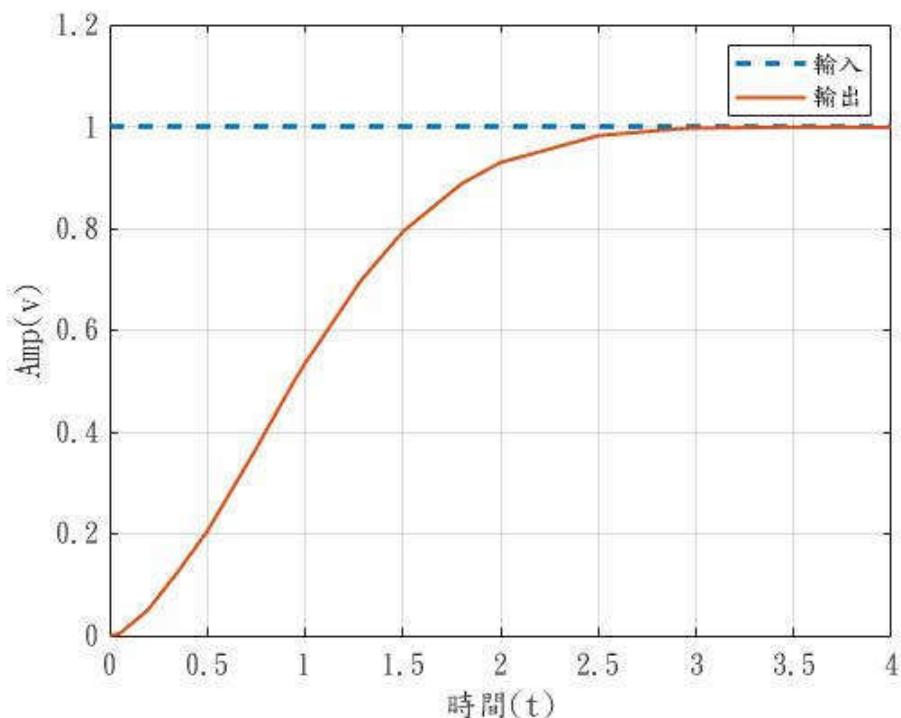
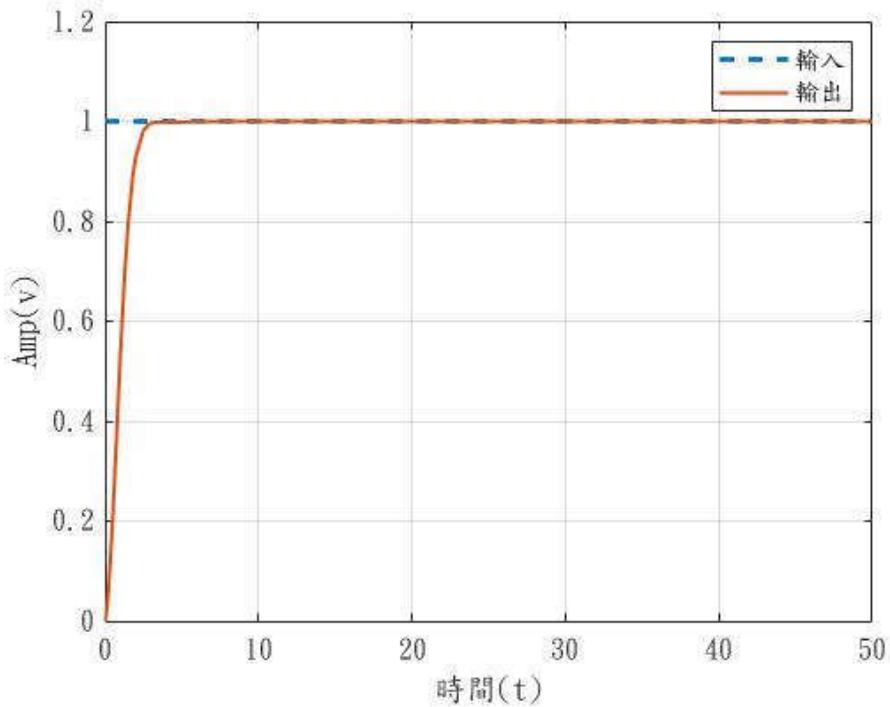
N_e : 5.101623e-01

N_{ce} : 4.416708e-01

N_u : 5.611612e-02

3. 兩張圖

i. 輸出輸入同一張圖



上圖時間單位為秒，系統可以達到大約三秒左右進入安定，最大超越量小於30%，穩態誤差也接近零。

ii. 控制訊號

